

Position and Obstacle Avoidance Algorithm in Robot Soccer

Awang Hendianto Pratomo, Anton Satria Prabuwono, Mohd. Shanudin Zakaria, Khairuddin Omar, Md. Jan Nordin, Shahnorbanun Sahran, Siti Norul Huda Sheikh Abdullah and Anton Heryanto
Center for Artificial Intelligence Technology, Faculty of Information Science and Technology,
University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor DE, Malaysia

Abstract: **Problem statement:** Robot soccer is an attractive domain for researchers and students working in the field of autonomous robots. However developing (coding, testing and debugging) robots for such domain is a rather complex task. **Approach:** This study concentrated on developing position and obstacle avoidance algorithm in robot soccer. This part is responsible for realizing soccer skills such as movement, shoot and goal keeping. The formulation of position and obstacle avoidance was based on mathematical approach. This formula is to make sure that the movement of the robot is valid. Velocity of the robot was calculated to set the speed of the robot. The positioning theory including the coordination of the robot (x,y) was used to find the obstacle and avoid it. **Results:** Some simulations and testing had been carried out to evaluate the usefulness of the proposed algorithms. The functions for shooting, movement and obstacle avoidance had been successfully implemented. **Conclusion:** The results showed its possibility could be used as strategy algorithms in real robot soccer competition.

Key words: Position algorithm, obstacle avoidance algorithm, robot soccer strategy, autonomous robot

INTRODUCTION

The idea of robot soccer was born in 1995 in the Korea Advanced Institute of Science and Technology (Chhabra *et al.*, 2004). Robot soccer is one of the research areas in autonomous robot systems. It is a five on five soccer competition between small, fully autonomous robots. There are different robots that have to work together toward a common goal. The domain is continuous and dynamic. There are opponents whose behavior will not be fully predictable. Because of the competitive element of the game, it is necessary to act sensible and fast. This together with the fact that the game offers a constricted controllable domain and is entertaining and challenging makes it an ideal test-bed for multi-agent collaborating robotics researches.

However, in order to keep the game as close as possible to the real game of soccer, most of rules used in human soccer are also used in robot soccer. To achieve the goal of an autonomous team of soccer playing robots, various technologies have to be incorporated including control theory, distributed systems, computer vision, machine learning, communication, sensor data fusion and self localization and team strategies. In order to do researches at as many different levels as possible several different

leagues exist (Groen and Vlassis, 2002). This research refers to the small-size robot league.

Basic concept of robot soccer: Robot soccer is the small-size league played on a table-tennis sized field. Each team consists of five small robots. A camera above the field is used to get a complete view of the game, which is send to the computers of the teams on the side of the field. From this image a world model is constructed using the color coding of the ball and the different robots. We use an orange golf ball as the soccer ball. Using this world model the actions of the different robots are determined and send to the robots. The games in this league are typically very fast and chaotic.

There are five main subsystems in which all others are contained. Fig. 1 shows a high-level system block diagram of autonomous robot systems:

- The vision system records data on the robot and obstacle positions
- The strategy system decides where the robots should move and solve the problem
- The communication system transmits the commands from the CPU to the robot, in this research using Bluetooth

Corresponding Author: Anton Satria Prabuwono, Center for Artificial Intelligence Technology,
Faculty of Information Science and Technology, University Kebangsaan Malaysia, 43600 UKM
Bangi, Selangor DE, Malaysia

- The on-board control system processes the data received from the communication system and executes the required commands
- The robot system controls the mechanical elements of the robot (Novak, 2002)

Fig. 2 shows a configuration of autonomous robot systems. The first part of system is the vision system. Vision systems use an overhead camera, which is mounted 2.8 m above the playing field. The camera is used to capture images from the field and relays the information to computer systems. The data capture is used to sort out colors and determine color blobs from the captured image. The vision system used to identify and decipher the different robots and each individual's orientation. The vision system takes a real-time image from an overhead camera and processes that image to determine the position and orientation of the objects on the field. This task is extremely processor intensive and must be optimized to utilize the processor in an expeditious manner.

Strategy system allows the robots to initiate strategies relative to the position of other robots and obstacle on the field. That system received the data from vision system and processes it through a calculated to determine which type of strategy should be issued. The goal is to provide strategic functionality to the robots and create different algorithms to suit different scenarios that our robots might to finish the task (Novak, 2002).

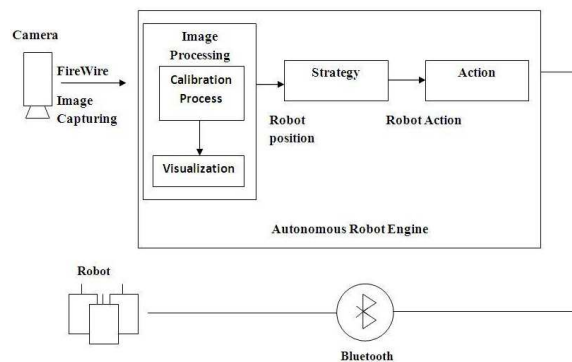


Fig. 1: Block diagram of autonomous robot systems

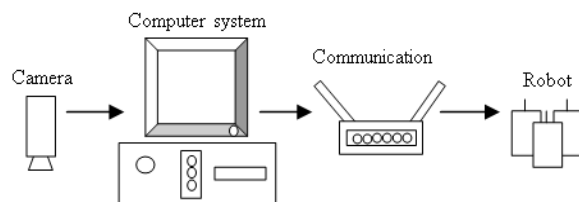


Fig. 2: Systems configuration

Communication and on-board control system received data from the strategy system using wireless communication from the computer to the robots. This subsystem processed the values given in the protocol from the strategy system to determine distance, direction and degree of orientation. This system is given data from the vision system and through the strategy system and the finally to the communication system for the robots to execute actual movement. The data is updated continuously through the vision and strategy system so the communication system can update and its movements effectively in real-time. Robots execute the instructions such as stop, move and rotate and their positions are tracked using the overhead camera. Robotic mechanisms will drive the robot and complete the task (Novak, 2002).

The first design of the movement focuses on standalone robot without any collision. The robot only uses their sensor with the data captured by the camera and moves as desired. The robot generates desired velocities to move itself from the initial point to the target point without collision. At each time step, the necessary input data are received in real time from the vision system. Then the robot generates the reference linear and angular velocity to the target point.

In order to generate the reference linear and angular velocity, the robots take the input which the data includes distance and angle between obstacles. We take the value of 1) the distance to the goal 2) the distance to the closest obstacle 3) the angle between the current robot orientation and the direction to the goal 4) the angle between the current robot orientation and the direction to the closest obstacle (Kim *et al.*, 2007). We concentrate on developing the function of robot soccer including how the robot avoids the obstacle around it.

MATERIALS AND METHODS

Each team of mobile robot is equipped with a camera that located in above the playing field. The image data are read from an image-processing module that must try to detect an object in the image using image-processing algorithm (sensing). The object is analyzed and reconstructed after a successful detection. The information then been extract and interpret. It is necessary to determine its world coordinates. The three-dimensional world coordinate system is independent of the robot's actual position. Its origin can be arbitrarily chosen. For example, the origin could be that point from which a robot starts its interior exploration (Kirylov, 2006).

The origin of the three-dimensional camera coordinate system is determined by the focal point of

the camera. If object coordinates are actually known in the camera coordinate system, it is possible to derive the world coordinates. After the three-dimensional object reconstruction is completed, the examined data can be collected in the navigation map. This can be said as a localization map building.

The determination of the coordinates can use a stereo technique. At least two images from different positions are necessary for these purposes. Corresponding pixels belonging to that image region, which represents the desired object, must be detected in both images. Stereo triangulation exploits geometrical realities to determine the distance of the object point from the focal point. Additionally, the technical data of the camera must be considered for the depth estimation.

Fig. 3 shows the architecture and control schema of robot soccer. The program determines for every edge the length and its start and endpoints, which are represented by nodes. Coordinates are then attached to every node. Next step is the cognition path planning. After the robot has detected all the coordination and by using the information it gain, it will proceed to the planning on where it should move. The robot will go through the algorithm that has been set and then it choose the best solution and execute the movement. After that, the order is sent to the robot in the real world for real movement (Baharin, 2009).

Simulator: Fig. 4 shows the version 1.5 Robot Soccer Simulator. An Australian teacher, Dr. Jun Jo created this simulator (Kim *et al.*, 2007). This is the Federation of International Robot-soccer Association (FIRA) official simulator. It is used to test the strategy before implements to the real robot. To do some tests, we use the software of Microsoft Visual C++ 2008®. In the code, we produce a file DLL, which is copied in the folder strategy of the simulator. The strategy property associated with each player can be used to point to a DLL which implements the strategy interface.

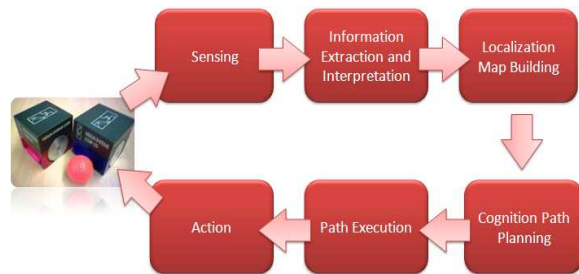


Fig. 3: Architecture and control schema of robot soccer

For example there is a goalkeeper DLL that determines the behavior of the goalie and another DLL that is specific to a striker. In function of the tactical schemas developed, there are two others DLL with wingers. All mathematic functions can be used in the program for all the applications (Krylov, 2006).

Formulation: Understanding the concept is the most important thing in designing the movement of the robot. Mathematics formula is used in most applications to make sure that the movement of the robot is valid. Velocity of the robot is calculated as to set the speed of the robot. It related most with the angle error (Kim *et al.*, 1998). The relationship is depending on proportional gain, K_p where is the proportional refer to a correction signal creation which is proportional to the error. This way, we can budget the turning degrees in which also including the measure of the distance for all directions. The formula to calculate the velocity as below:

$$V_L = K_1 \cdot d_e - K_a \cdot \theta_c \quad (1)$$

$$V_R = K_1 \cdot d_e - K_a \cdot \theta_c \quad (2)$$

where, K_p is a proportional gain

K_p is set differently for different angle error. This is due to the relationship between angle errors and velocity. If smaller K_p is used for bigger angle errors and also for the small angle error, the velocity values sent to the robot are smaller than the required values.

Algorithm: The basis for a lot of functionality was the skills that move a robot to a position or pose (Jolly *et al.*, 2009; Siegwart and Nourbakhsh, 2004). At the moment we do not consider any obstacles. In this case, we apply the positioning theory as shown in Fig. 5.



Fig. 4: FIRA simulator

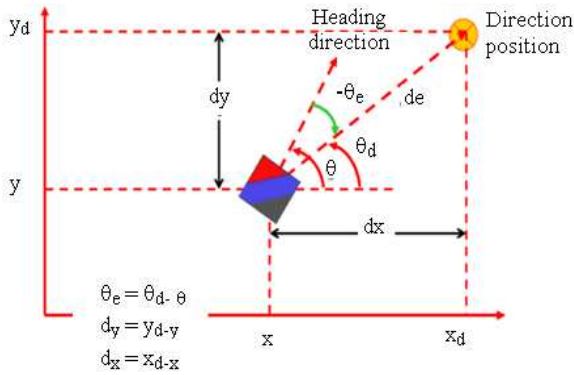


Fig. 5: Positioning theory

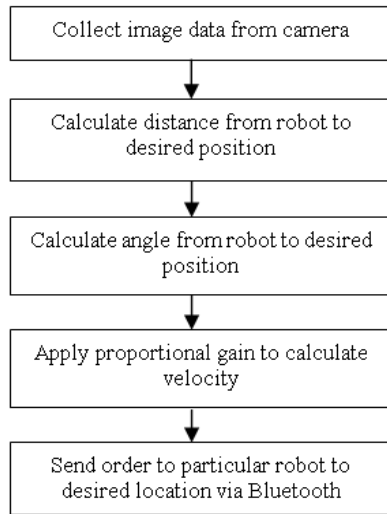


Fig. 6: Position algorithm

It is similar with the angle error. It includes the coordination of the robot (x,y). This theory is the same application to find distance within a triangle shape (Baharin, 2009).

There are also a relationship between distance angle error and velocity. Still in this case, the relationship is depending on proportional gain. If the smaller proportional gain that is used for bigger and small distance error, the velocity data values sent to the robot are smaller than required values. Thus, for different situation we set the different gain (Egley *et al.*, 2005).

Fig. 6 shows the position algorithm that developed in this research.

Furthermore the algorithm for obstacle avoidance that developed in this research is shown in Fig. 7.

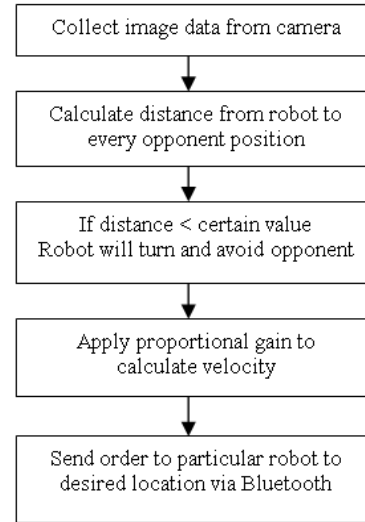


Fig. 7: Obstacle avoidance algorithm

The codes for positioning algorithm are shown in the following codes.

```

void Position( Robot *robot, double x, double y )
{
    int desired_angle = 0, theta_e = 0, d_angle = 0,
    vl, vr, vc = 70;
    double dx, dy, d_e, Ka = 10.0/90.0;
    dx = x - robot->pos.x;
    dy = y - robot->pos.y;

    d_e = sqrt(dx * dx + dy * dy);
    if (dx == 0 && dy == 0)
        desired_angle = 90;
    else
        desired_angle = (int)(180. / PI *
        atan2((double)(dy), (double)(dx)));
    theta_e = desired_angle - (int)robot->rotation;

    while (theta_e > 180) theta_e -= 360;
    while (theta_e < -180) theta_e += 360;
    if (d_e > 100.)
        Ka = 17. / 90.;
    else if (d_e > 50)
        Ka = 19. / 90.;
    else if (d_e > 30)
        Ka = 21. / 90.;
    else if (d_e > 20)
        Ka = 23. / 90.;
    else
        Ka = 25. / 90.;

    if (theta_e > 95 || theta_e < -95)
  
```

```

{
    theta_e += 180;

    if (theta_e > 180)
        theta_e -= 360;
    if (theta_e > 80)
        theta_e = 80;
    if (theta_e < -80)
        theta_e = -80;
    if (d_e < 5.0 && abs(theta_e) < 40)
        Ka = 0.1;
        vr = (int)(-vc * (1.0 / (1.0 + exp(-3.0 *
d_e)) - 0.3) + Ka * theta_e);
        vl = (int)(-vc * (1.0 / (1.0 + exp(-3.0 *
d_e)) - 0.3) - Ka * theta_e);
    }

    else if (theta_e < 85 && theta_e > -85)
    {
        if (d_e < 5.0 && abs(theta_e) < 40)
            Ka = 0.1;
            vr = (int)(vc * (1.0 / (1.0 + exp(-3.0 *
d_e)) - 0.3) + Ka * theta_e);
            vl = (int)(vc * (1.0 / (1.0 + exp(-3.0 *
d_e)) - 0.3) - Ka * theta_e);
        }

        else
        {
            vr = (int)(+.17 * theta_e);
            vl = (int)(-.17 * theta_e);
        }

        Velocity ( robot, vl, vr );
    }

    Then the codes for obstacle avoidance are shown in
    the following codes:

    double ObstacleAvoidace( double x, double y, double
ox, double oy, double ro, double m, double theta_d )
    {
        double dist, length, angle, diff_angle;
        double tmp_x, tmp_y;

        // distance between robot and obstacle
        dist = sqrt((ox-x)*(ox-x) + (y-oy)*(y-oy));
        //
        length = fabs( (ox-x)*sin(theta_d) + (y-
oy)*cos(theta_d) );
        angle = atan2( oy-y, ox-x );
        diff_angle = theta_d - angle;

        while( diff_angle > PI ) diff_angle -= 2.*PI;

        while( diff_angle < -PI ) diff_angle += 2.*PI;

        if( (length < ro+m) && (fabs(diff_angle) < PI/2))
        {
            if( dist <= ro ) theta_d = angle - PI;
            else if( dist <= ro+m )
            {
                // modify theta_d to avoid it with CW direction
                if( diff_angle > 0. )
                {
                    // make smooth transition near the obstacle
                    //boundary
                    tmp_x = ( (dist-ro)*cos(angle-1.5*PI) +
(ro+m-dist)*cos(angle-PI) ) / m;
                    tmp_y = ( (dist-ro)*sin(angle-1.5*PI) +
(ro+m-dist)*sin(angle-PI) ) / m;
                    theta_d = atan2( tmp_y, tmp_x );
                }
                // modify theta_d to avoid it with CCW
                // direction
                else
                {
                    // make smooth transition near the obstacle
                    // boundary
                    tmp_x = ( (dist-ro)*cos(angle-0.5*PI) +
(ro+m-dist)*cos(angle-PI) ) / m;
                    tmp_y = ( (dist-ro)*sin(angle-0.5*PI) +
(ro+m-dist)*sin(angle-PI) ) / m;
                    theta_d = atan2( tmp_y, tmp_x );
                }
            }
            else
            {
                // modify theta_d to avoid it with CW direction
                if( diff_angle > 0. )
                {
                    theta_d = fabs( atan( (ro+m) / sqrt(
dist*dist - (ro+m)*(ro+m) ))) + angle;
                }
                // modify theta_d to avoid it with CCW
                // direction
                else
                {
                    theta_d = -fabs( atan( (ro+m) / sqrt(
dist*dist - (ro+m)*(ro+m) ))) + angle;
                }
            }
        }

        return theta_d;
    }
}

```

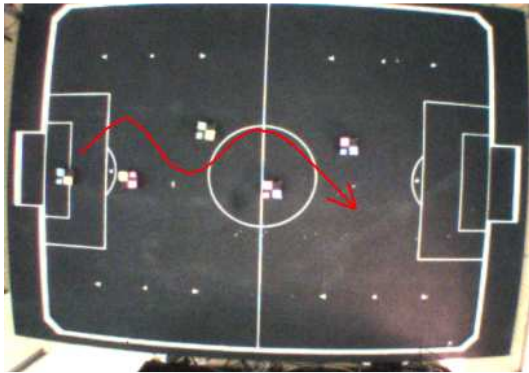


Fig. 8: Robot movement in curve shape



Fig. 9: Robot goes straight to opponent goalie

The robot is set to move to the others side field. Since there are obstacles throughout the path, the robot will take a movement in curve shape so that it will not collide with the other obstacle/robot as shown in Fig. 8. This movement can be successfully tested with the calculation of the angle between the robot, desired position and the obstacle.

The robot is set to move to the opponent goal. Obstacle is put in a straight way as shown in Fig. 9. The robot will now make a move in straight line to avoid collide with the other robots. Based on the calculation of the angle, the robot will move in curve in such degrees it should be.

DISCUSSION

In this research the algorithms for obstacle avoidance and position control in robot soccer have been developed. Mathematics formulation and positioning theory were implemented in order to get the valid experiment. Velocity of the robot is calculated to set the speed of the robot. It related most with the angle

error. The relationship is depending on proportional gain, K_p where is the proportional refer to a correction signal creation which is proportional to the error. We can arrange the turning degrees in which also including the measure of the distance for all directions.

CONCLUSION

The general formulation of basic movement of the robots has been introduced through several examples. The formulation most based on mathematical approach. The basic movements in robot soccer have been presented as well. The functions or movements that successfully developed are shooting, movement, blocking and goal keeper. Microsoft Visual C++ 2008® was used to implement the algorithms. Finally, the basic function algorithms were successfully implemented and tested by using Robot Soccer Simulator V1.5A and lively tested using the real robots. The results show its possibility could be used as strategy algorithms in real robot soccer competition.

ACKNOWLEDGMENT

The researchers would like to thank Faculty of Information Science and Technology, University Kebangsaan Malaysia for providing facilities and financial support under Research University Grant of Pattern Recognition Research Group No. UKM-GUP-TMK-07-02-034. Besides, thanks to Beh Kheng Aik, Noor Izzati Mohd Nasir, Lai Yi Qing and Ruzaini for their contribution in this research.

REFERENCES

- Baharin, K.N., 2009. Robotic soccer programming. Astana Digital paper sheet. <http://www.astanadigital.com/>
- Chhabra, M., A. Nahar, A. Mukherjee, A. Mathad and S. Chaudhuri, 2004. Novel approaches to vision and motion control for robot soccer. Proceedings of the National Conference on Advanced Manufacturing and Robotics, (AMR'04), Allied Publishers Pvt. Ltd., India, pp: 68-74.
- Egly, U., G. Novak and D. Weber, 2005. Decision making for miroSOT soccer playing robots. http://www.tinyphoon.com/rainbow/_tinyphoon/Documents/CLAWAR_EURON_DecisionMaking.pdf
- Groen, F., M. Spaan and N. Vlassis, 2002. Robot soccer: Game or science. Proceedings CNR-2002, (CNR'02), Editura Universitaria Craiova, Romania, pp: 92-98.

- Jolly, K.G., S. Kumar and R. Vijayakumar, 2009. A bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits. *Robot. Auton. Syst.*, 57: 23-33. DOI: 10.1016/j.robot.2008.03.009
- Kim, C.J., M.S. Park, A.V. Topalov, D. Chwa and S.K. Hong, 2007. Unifying strategies of obstacle avoidance and shooting for soccer robot systems. *Proceedings of the International Conference on Control, Automation and Systems*, Oct. 17-20, IEEE Xplore Press, Seoul, Korea, pp: 207-211. DOI: 10.1109/ICCAS.2007.4406909
- Kim, J.H., K.C. Kim, D.H. Kim, Y.J. Kim and P. Vadakkepat, 1998. Path planning and role selection mechanism for soccer robots. *Proceedings of the International Conference on Robotics and Automation*, May 16-20, IEEE Xplore Press, Leuven, Belgium, pp: 3216-3221. DOI: 10.1109/ROBOT.1998.680920
- Kyrylov, V., 2006. Balancing Gains, Risks, Costs and Real-Time Constraints in the Ball Passing Algorithm for the Robotic Soccer. *Lecture Notes Comput. Sci.*, 4434: 304-313.
- Novak, G., 2002. Multi agent systems-robot soccer. Thesis, Vienna University of Technology, Vienna, Austria. <http://www.tuwien.ac.at>
- Siegwart, R. and I.R. Nourbakhsh, 2004. *Introduction to Autonomous Mobile Robot*. 1st Edn., MIT Press Cambridge, Massachusetts, USA, ISBN: 026219502X, pp: 10-12.